

## **Random neural network based cognitive engines for adaptive modulation and coding in LTE downlink systems**

Adeel, Ahsan; Larijani, Hadi; Ahmadinia, Ali

*Published in:*  
Computers and Electrical Engineering

*DOI:*  
[10.1016/j.compeleceng.2016.11.005](https://doi.org/10.1016/j.compeleceng.2016.11.005)

*Publication date:*  
2017

*Document Version*  
Author accepted manuscript

[Link to publication in ResearchOnline](#)

*Citation for published version (Harvard):*  
Adeel, A, Larijani, H & Ahmadinia, A 2017, 'Random neural network based cognitive engines for adaptive modulation and coding in LTE downlink systems', *Computers and Electrical Engineering*, vol. 57, pp. 336-350.  
<https://doi.org/10.1016/j.compeleceng.2016.11.005>

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

### **Take down policy**

If you believe that this document breaches copyright please view our takedown policy at <https://edshare.gcu.ac.uk/id/eprint/5179> for details of how to contact us.

# Random Neural Network based Cognitive Engines for Adaptive Modulation and Coding in LTE Downlink Systems

Ahsan Adeel, Hadi Larijani\*, Ali Ahmadinia

*School of Engineering & Built Environment, Glasgow Caledonian University, UK  
Computer Science and Information Systems Department, California State University San Marcos, US*

---

## Abstract

This paper presents two random neural network (RNN) based context-aware decision making frameworks to improve adaptive modulation and coding (AMC) in long-term evolution (LTE) downlink systems. In the first framework, AMC is modelled as a traditional classification problem with the aim to maximize the probability of correct classification. The second framework seeks to optimize the throughput as opposed to simply maximizing the probability of the correct classification. To model the second framework, we developed a hybrid cognitive engine (CE) architecture by integrating an RNN based learning algorithm with genetic algorithm (GA) based reasoning. RNN features help CE to concurrently acquire long-term-learning, fast decision making, and less computational complexity, which are essential for the development of any real-time cognitive communication system. The performance of RNN is compared with artificial neural networks (ANN) and state-of-the-art effective exponential SINR mapping (EESM) algorithm. A comprehensive analysis of the proposed RNN based AMC scheme is presented by jointly incorporating the effect of different schedulers, feedback delays, and multi-antenna diversity on the throughput of an orthogonal frequency-division multiple access (OFDMA) system. The critical analysis of the first framework revealed that RNN based CE can achieve comparable results with faster adaptation, even in severe environment changes without the need of retraining compared to ANN. The analysis of the second approach demonstrated RNNs faster adaptation as compared to ANN with up-to 100% improvement in both system capacity and fairness as compared to EESM algorithm.

**Keywords:** Random neural network, context-aware decision making, adaptive modulation and coding, long-term evolution, hybrid cognitive engine, genetic algorithm

---

## 1. Introduction

To achieve the desired spectral efficiency, the 3G, 4G, and beyond have employed adaptive approaches to dynamically adapt radio configuration parameters. These techniques are generally known as link adaptation or channel-aware scheduling. One important method for doing so is AMC. AMC aims to maximize the data rate by selecting an optimal modulation and coding scheme (MCS) under block-error-rate (BLER) reliability constraint. According to the 3rd generation partnership project (3GPP) specification [1], user equipment (UE) periodically reports its channel state information (CSI) to the base station (BS). Based on this feedback, BS performs the process of AMC.

In an orthogonal-frequency-division multiplexing (OFDM) system, a transport block (TP) is encoded over several sub-carriers. Therefore, AMC can't be done for individual sub-carriers, because it proportionally increases the control and signalling overhead. This difficulty motivates the extensive use of EESM technique. The EESM technique translates signal-to-interference-and-noise ratios (SINR) of multiple sub-carriers into an effective one-dimensional SINR ( $\gamma_{eff}$ ). Based on  $\gamma_{eff}$ , same MCS for all assigned sub-carriers (sub-band) to each UE is selected [2][3]. The  $\gamma_{eff}$  can be calculated by performing a following non-linear averaging over several sub-carriers SINRs:

---

\*Corresponding author

Email address: H.Larijani@gcu.ac.uk (Hadi Larijani)

$$\gamma_{eff} = EESM(\gamma_i, \beta) = -\beta \cdot \ln\left(\frac{1}{N} \cdot \sum_{i=1}^N \exp^{-\frac{SINR_i}{\beta}}\right) \quad (1)$$

where  $N$  is the total number of subcarriers to be averaged and  $\beta$  is calibrated to fit the compression function to the Additive White Gaussian Noise BLER results [4]. The UE uses EESM to compute the  $\gamma_{eff}$ , which represents the channel quality. However, the reduction of  $N$  subcarriers SINRs to a single effective SINR is not information lossless that needs to be dealt with the rate adaptation and scheduling.

It is to be noted that (1) is a non-linear function and no exact closed-form expression for its statistics is known [5][6]. The difficulty in constructing an exact closed-form expression is due to the high dimensions of transmission parameters (e.g. frequency, time, spatial domain) and environmental parameters (e.g. signal energy, noise variance, channel variations per subcarrier, time tap etc.) These multi-dimensions makes it almost impossible to come up with a closed form mapping of environmental measurements to transmission parameters. Moreover, the factors such as non-linearities of system, quantization errors, and non-Gaussian noise adds more to this difficulty and correspondingly the selection of optimal or even near optimal MCS. Therefore, for effective link adaptation, a flexible framework is desired to enable AMC using as few assumptions on the mathematical model of the physical layer as possible.

Recently, machine learning (ML) based link adaptation approaches have been extensively studied in literature. ML translates AMC in to a decision making process, because for ML AMC is nothing more than a transfer between data observation and system state [7]. ML algorithms makes no mathematical assumptions and learn the input-output relationship using the training process. Therefore, ML based AMC approaches have better ability to capture the environmental effects as compared to the classical EESM method. In addition, ML based AMC approaches are capable to address the limitations of analytical modelling such as lack of ability to deal with non-ideal communication behaviours, poor scalability, limited modelling assumptions etc.

Application of machine learning to AMC has been studied widely using both supervised and unsupervised learning approaches such as ANN [8], k-nearest neighbours (k-nn) [7], support vector machine (SVM) [9], reinforcement learning (RL) [10] etc. However, none of the existing ML based CE design fully complies with the CE design requirement i.e. the concurrent achievement of long-term learning, fast decision making, and less computational complexity. In addition, most of the proposed ML based approaches have translated AMC as a classification problem.

In this paper, we utilize advanced artificial intelligence (AI)/ML techniques such as RNN to address the limitations of existing ML based AMC approaches. In our previous work [11], [12], and [13], we addressed LTE uplink power control problem using RNN. The use of RNN in [11][12][13] helped CEs to comply fully with the essential CE design requirements. Consequently, it was proved that RNNs inherent properties such as: (a) easy and efficient computation (b) low complexity of standard learning algorithm (c) energy-efficient hardware implementation (d) less dependence on network structure (e) strong generalization capability even with small training dataset, makes RNN a better choice for CE design [14]. Therefore, this paper extends our previous research by presenting RNNs first application to the problem of link adaptation in LTE downlink.

In the rest of the paper, Section 2, 3, and 4 briefly describes the related work, motivations, and contributions respectively. Section 5 presents an overview of RNN, its mathematical modelling, and advantages. Section 6 describes the system model. Section 7 has presented the proposed CE design 1 and its evaluation. Similarly, Section 8 presents the proposed CE design 2 and its evaluation. Finally, Section 9 has concluded this work.

## 2. Related Work

Motivated by the goal of analysing the channel quality information (CQI) feedback, the authors in [5] proposed the use of an analytically traceable logarithmic distribution to characterize the probability distribution function of  $\gamma_{eff}$  in a Rayleigh channel distribution. Similarly, the authors in [6] proposed a novel statistical model for EESM based on the Beta distribution. However, these solutions are based on sound theoretical principles and their implementation require large amount of calculations, which supposed to be calculated at very fine-grained time intervals. In addition, these schemes suffers from the limitations of analytical modelling as highlighted in Section 1 [7].

The authors in [8] presented an ANN based AMC process for multi-input-multi-output (MIMO)-OFDM wireless system using Levenberg-Marquardt (LM) training algorithm. However, ANNs suffers from training and local-minima

problems, high computational complexity, limited generalization, slow calculation rate during run-time, and sensitivity to the number of hidden neurons/available training samples. The authors in [7] proposed a k-NN based AMC approach to exploit the past observation of CSI and predict the best MCS. This approach provided accurate mapping from features to MCS and outperformed other compared link adaptation algorithms. However, the practical application of this approach is limited by excessive processing and memory requirement. Moreover, the k-NN algorithm requires all previous samples to be stored in order to predict future MCS, which is simply too expensive.

The authors in [9] and [15] extended the work of [7] and developed two new AMC processes based on SVMs. The approach outperformed k-NN based adaptation algorithm in terms of better spectral efficiency, less time-overhead, and significant reduction in memory requirement. However, SVM based approaches have some important practical questions such as the selection of kernel function and its parameters, training of multi-class SVM, high algorithmic complexity, extensive memory requirement, and slow speed during run-time [16].

The authors in [10] addressed the time complexity and offline training issues of their previous work [15] by using RL. The RL based online AMC approach showed comparable performance as compared to the existing online approaches such as [9] but with minimal time and storage requirement. However, for realistic applications, the size of state-space could be so large that learning may take a long time and even become impossible in a reasonable time frame. As a result, the generalization over the state-space is necessary, which is insufficient in RL.

### 3. Motivation

The key required feature for the development of ML based systems is to have an effective learning capability. ML based systems learn the desired system behaviour through the process of training and the choice of any training algorithm exhibits a trade-off among available training samples, accurate learning, training speed, and computational complexity. In literature, several methods are proposed to train CE in a reasonable amount of time and effort [17]. However, very little effort has been made to reduce the need of retraining upon severe wireless environment changes, which are essential, because retraining consumes valuable time and energy. Moreover, when CE is working on a critical mission without any time to retrain, then, retraining is definitely not an option. There are two possible ways to avoid retraining: one is to put the CE through all expected operating conditions during training phase and second is to attain long-term learning capability. The first solution is practically impossible to explore all possible wireless conditions *a priori*. Therefore, it is reasonable to assume that CE might face an unknown condition, which requires retraining [17]. Therefore, the capability of long-term learning is of great importance which enable CEs to adapt themselves with respect to severe change in environment without the need of retraining. Insufficiency of long-term learning along with the concurrent achievement of fast decision making and less complexity are the major limiting factors of existing ML based AMC approaches. The concurrent achievement of these three features is essential for the development of any successful cognitive communication system. In addition, for capacity optimization, AMC process should be modelled as an optimization problem.

### 4. Contributions

The main contributions of this paper are:

1. Development of an RNN based CE for AMC in LTE multi-cell/multi-user scenario. The framework translated AMC into a traditional classification problem with the focus to maximize the probability of correct classification. The developed CE complies with the essential CE design requirement and avoids the need of retraining upon severe electromagnetic environment change.
2. Extended contribution number 1 and previously proposed ML based AMC approaches such as [7][9]. The MCS selection is modeled as an optimization problem with the aim to maximize the throughput as opposed to maximize the probability of correct classification. To achieve optimization, we developed a hybrid CE architecture based on RNN based learning and GA based reasoning.
3. Comprehensive and critical analysis of the proposed scheme by jointly incorporating the effect of scheduler, feedback delay, multi-antenna diversity, and EESM on the throughput of an OFDMA system, which covers a wide range of coverage and capacity tradeoff.

## 5. Random Neural Network

RNN, a machine learning technique, made up of highly interconnected processing elements called as neurons, processes the information by their state response and learn from examples. The main objective of the RNN model is to transform the inputs into meaningful outputs, learn the input-output relationship, and offer viable solutions to unseen problems (a generalization capability). RNNs were first developed by Gelenbe in [18] as a new modified class of ANNs, representing the transmission of signals in a very similar form to biological neural networks, but offers more benefits and cope the limitations of ANNs. In RNN, a neuron can be seen as a queue with an exponential server having (service) rate  $\mu$  and exciting/inhibition signals as positive/negative customers. In case of no inhibition signal, RNN behaves as a classic M/M/1 queue.

### 5.1. Mathematical Modelling

In RNN, the neuron exchanges the signal in the form of spikes. The potential ( $k$ ) of each neuron represents its state that increases/decreases with respect to an incoming signal. A neuron  $u$  can receive exogenous signals positive/negative, modelled as Poisson arrival streams of rates  $\Lambda_u, \lambda_u$ , respectively. If a neuron receives an excitatory signal (+1), its potential increases and correspondingly decreases upon receiving inhibitory signal (-1). When the potential of neuron is equal to zero ( $k_i = 0$ ), it is in idle state and when ( $k_i > 0$ ), the neuron is excited. In the state of excitation, the neuron fires an excitatory spike that goes from neuron  $u$  to  $v$  of the network or to the outside world. In that case, the potential of neuron  $u$  decreases by one, whereas potential of neuron  $v$  increases by one. When neuron fires inhibitory spike, the potential of both neuron decreases by one. This firing is according to the Poisson process represented by the synaptic weights  $w_{ij}^+ = rP_{ij}^+$  and  $w_{ij}^- = rP_{ij}^-$ , where  $P_{ij}^+$  and  $P_{ij}^-$  are the probabilities of excitatory and inhibitory signals and  $r$  is the spikes firing rate. The  $w_{ij}^+$  and  $w_{ij}^-$  can be seen as the positive and negative rates of signal transmissions and these are the typical interconnections weights of a neural network that RNN learns through the process of learning or training. The average rate of +ive signals at neuron  $i$  ( $\lambda_i^+$ ), average rate of -ive signals at neuron  $i$  ( $\lambda_i^-$ ), and the probability that neuron  $i$  is excited ( $q_i$ ), are calculated using the following equations:

$$\lambda_i^+ = \Lambda_i + \sum_{j=1}^n q_j w_{ij}^+ \quad (2)$$

$$\lambda_i^- = \lambda_i + \sum_{j=1}^n q_j w_{ij}^- \quad (3)$$

$$q_i = \frac{\lambda_i^+}{r_i + \lambda_i^-} \quad (4)$$

#### 5.1.1. Network Behaviour in Steady State

if  $0 \leq q_i \leq 1$  for  $i=1,2,3,\dots,n$  then the stationary joint probability of network  $p(k, t) = p_r = [k(t) = k]$  can be written as:

$$p(k) = \prod_{i=1}^n (1 - q_i) q_i^k \quad (5)$$

#### 5.1.2. Network Stability

In [19], Gelenbe presented the mathematical usability and stability of RNN model. The author showed that network is stable if the potential of the signal increases with bounds, which can be guaranteed if a unique solution to non-linear equations (2-4) exists. However, existence of the solution implies its uniqueness and in feed-forward RNN, the solution always exists. A general RNN model is depicted in Figure 1. Further in-depth details are comprehensively presented in [19].

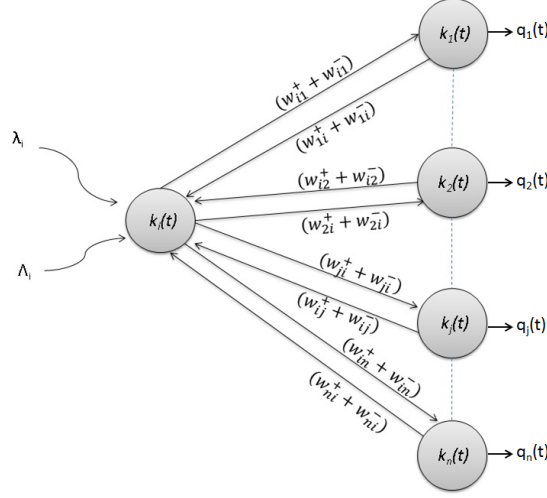


Figure 1: A feed-forward random neural network architecture

### 5.2. RNN Training Algorithms

The capacity to learn from examples is one of the most desirable features of neural network models. The goal of training is to learn desired system behaviour and adjust the network parameters (interconnections weights) to map (learn) the input-output relationship and minimize the mean squared error (MSE). In [20], Gelenbe presented a learning algorithm for the recurrent random network model using gradient descent (GD) of a quadratic error function. The learning algorithm introduced in [20] for recurrent RNN can also be applied to feed-forward RNN, which has been used in this paper. In [21][22], we also implemented adaptive inertia weight particle swarm optimization, differential evolution, and GA based learning algorithms. However, in general, there is a trade-off among learning accuracy, convergence time, calculation time, and computational complexity. Further details and procedures are presented in [20][21][22].

### 5.3. Random neural network distinguished features

- **Easy and Highly Energy-Efficient Implementation:** RNNs can be easily implemented in both hardware and software as its neurons can be represented by simple counters. A highly energy-efficient implementation of RNNs is demonstrated in [23] using the complementary metal-oxide semiconductor (CMOS) or Probabilistic CMOS (PCMOs) technology. The investigation showed RNNs 226-300 times more efficiency in terms of energy performance product as compared to ANNs. Furthermore, authors in [24] introduced a simplified RNN with  $2N$  weights instead of  $2N^2$  weights for the implementation of smart packet processing.
- **Low Complexity of Standard Learning Algorithm:** GD is a standard learning algorithm for RNN. GD has less computational complexity (i.e.  $O(N \times I \times H \times O)$ ) as compared to the non-standard training algorithms such as particle swarm optimization or differential evolution ( $O(N, I^2, H)$ ), where  $N$ ,  $I$ ,  $H$ , and  $O$  represents the number of training patterns, input, hidden, and output nodes respectively.
- **Easy and Efficient Computation:** The steady state probability distribution of RNNs can be described as analytical equations and can be efficiently computed without the use of Monte Carlo methods.
- **Strong Generalization Capability:** The authors in [25] compared the generalization capability of ANN and RNN for unseen patterns, which were not covered in the training dataset. The authors reported that RNN accurately classified the patterns while ANN failed to predict accurate output. In addition, RNNs showed better generalization even when they were trained with the small number of training data samples.
- **Less dependency on network structure:** Pedro Casas et al. in [26] showed that ANN performance is highly dependent on the number of neurons but RNNs dependency is not at the same degree. In addition, RNNs are more stable and accurate.

## 6. System Model

A schematic class diagram of the considered LTE system is depicted in Figure 2 [27][28]. The diagram is illustrating the whole LTE downlink and uplink operation. In addition, the diagram highlights the replaced EESM algorithm. The considered MIMO transmission mode is an open loop spatial multiplexing (OSLM) with  $N_t=N_r=2$  and the receiver design is simple zero forcing (ZF). At the transmitter side, each eNodeB constitutes three sectors and each sector consists a scheduler. The scheduler forms a resource grid and assigns precoding matrix, PHY resources, and MCS to each attached UE based on the received feedback. At the receiver side, the UE estimates the sub-carriers SINRs and calculate the CQI using EESM. The CQI is fed back to the scheduler via an uplink feedback channel with adjustable delay. Based on the CQI feedback, the scheduler performs the process of link adaptation by selecting an appropriate MCS. The CQI depends on the cell layout, shadow fading, large-scale macroscopic path-loss, and small scale microscopic fading. The system model has incorporated the effects of cell planning, schedulers, feedback delays, interference, and multi-antenna diversity. Implementation wise, the simulation is performed by defining a region of interest (ROI), where eNodeBs and UEs are positioned in a specified transmission time interval (TTI). Simulation parameters are presented in Table 1 and further modelling assumptions are discussed in the following sub-sections.

### 6.1. Modelling assumptions and calculations

To abstract the link quality for each eNodeB-UE pair, a per-subcarrier post equalization symbol SINR is used. The post equalization symbol SINR has taken into account three different propagation losses: (i) eNodeB dependent large-scale path-loss (macroscopic path-loss), (ii) site dependent shadow fading, and (iii) time dependent small-scale fading.

#### 6.1.1. Large-Scale Path-Loss and Shadow Fading

The large-scale path-loss is calculated according to [1], which has jointly modelled both the propagation path-loss and antenna gain. For the modelling of shadow fading, a low complexity 2D Gaussian process with appropriate spatial correlation is used, which is a variant of Cholesky decomposition [27][28].

#### 6.1.2. Time Dependent Small-Scale Fading

The calculation of time dependent small-scale fading depends on the precoding, MIMO channel matrix, and the receiver filter. The precoding for considered spatial multiplexing and receiver design with large-delay cyclic delay diversity (CDD) or large-delay CDD is given as [27][28]:

$$\begin{bmatrix} y_0(i) \\ \vdots \\ y_{(N_t-1)}(i) \end{bmatrix} = W(i) \cdot D(i) U \begin{bmatrix} x_0(i) \\ \vdots \\ x_{(v-1)}(i) \end{bmatrix} \quad (6)$$

Where  $N_t$  and  $v$  are the number of transmit antennas and number of layers respectively.  $D$  and  $U$  specifies the large-delay cyclic and delay diversity.

The received symbol is given as:

$$\hat{X} = (\hat{H}_0 F)^+ (H_0 F) X_0 + (\hat{H}_0 F^+ n) + \sum_{i=1}^{N_{int}} (\hat{H} F)^+ (H_i F) X_i \quad (7)$$

Where  $F, H_0, H_i$ , and  $\hat{H}$  represents the WDU matrix, intended signals' channel, interfering eNodeBs' channel, and estimated channel respectively. In addition, '+' denotes the pseudoinverse.

By denoting  $A = (\hat{H} F)^+ (H F)$ ,  $B = (\hat{H} F)^+$ ,  $C_i = (\hat{H}_0 F)^+ (H_i F)$ , and  $a_{ij} = A[i, j]$ , the  $SINR_i$  for the received symbol  $i$  can be expressed as:

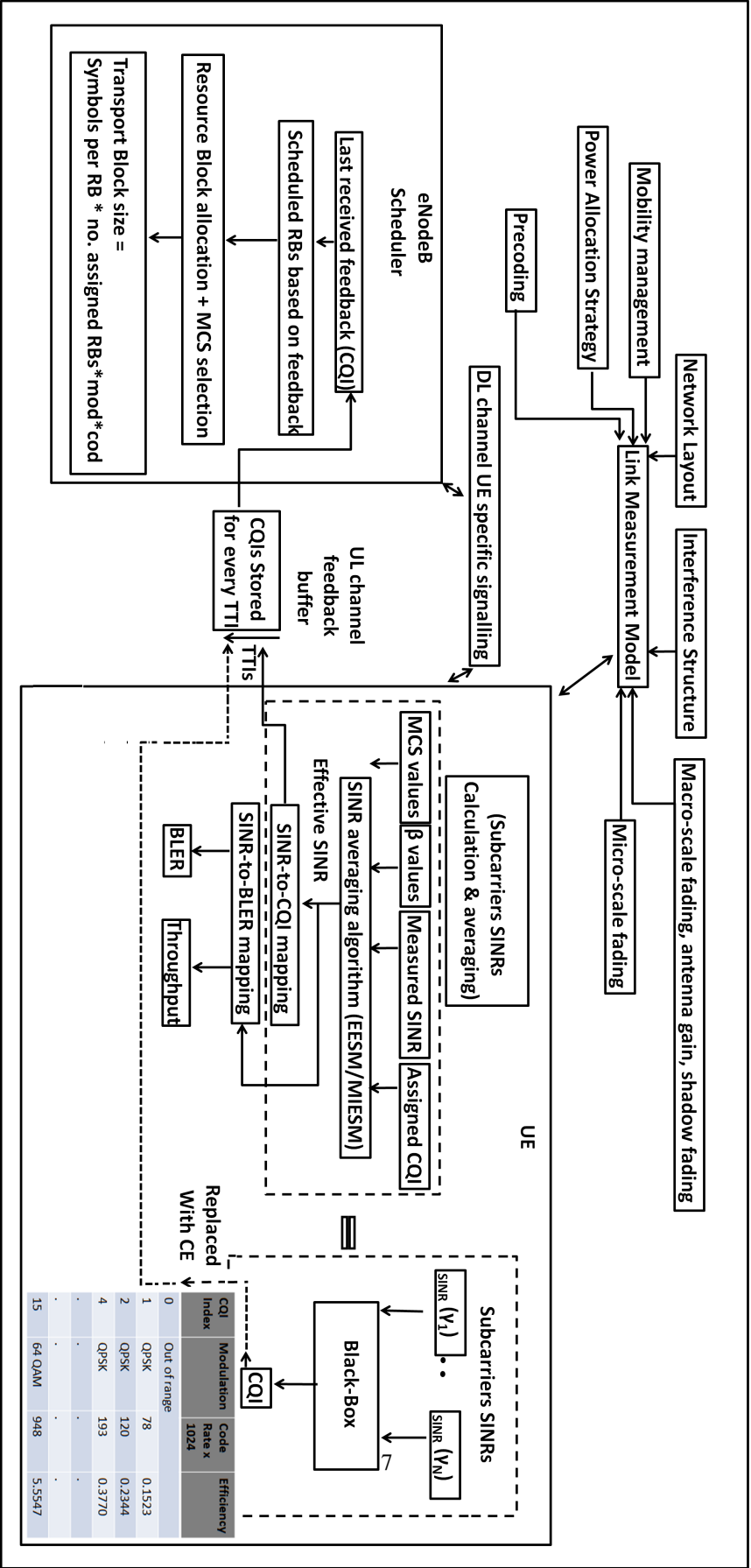


Figure 2: LTE system operation and replaced EESM averaging algorithm.



Table 1: Simulation Parameters

Parameters	Values
Network Topology	3 cells, 9 sectors
Frequency	$2e^9$ Hz
Bandwidth	$5e^6$ Hz
Number of antennas at transmitter	2
Number of antennas at receiver	2
Transmission mode	Open loop spatial multiplexing
Simulation time	100 TTIs
Number of TTIs used to calculate the average throughput (exponential filtering)	10
Inter eNodeB distance	500 meters
Minimum coupling loss	70 dB
Macroscopic path-loss model	TS25.814
eNodeB max transmission power	20 watts
Shadow fading type	Claussen
Receiver noise figure	9 dB
Receiver thermal noise density	-174 dBm/Hz
Total number of UEs	15
UE speed	15/3.6 Km/h
Antenna gain pattern	TS36.942
Antenna gain	15
Scheduler	Round robin, proportional fair, best CQI
Power allocation	Homogeneous
Uplink feedback channel delay	0, 1, 2, 3 TTIs
SINR averaging algorithm	EESM
RB bandwidth	$180e^3$ Hz
TTI length	$1e^{-3}$ seconds
Cyclic prefix	Normal
Maximum number of code words per TTI	2

$$SINR_i = \frac{|a_{ij}|P_i}{\sum_{j \neq i} P_j + \alpha^2 \sum_{k=1}^v |b_{ik}|^2 + \sum_{i=1}^{N_{int}} \sum_{m=1}^v |c_{i,m}|^2 P_{l,m}} \quad (8)$$

where  $P_i$  is the received power at  $i^{th}$  layer after macro and shadow fading losses and  $\alpha^2$  is the receiver noise. By assuming homogeneous power distribution ( $P_i = P_{Tx}/v$ ), channel estimation errors ( $\zeta, \xi$ ):  $\zeta = |a_{ii}|^2$ ,  $\xi = \sum_{j \neq i} |a_{ij}|^2$ , ZF receiver noise enhancement ( $\psi = \sum_{k=1}^v |b_{ik}|^2$ ), and the interference ( $\theta = \sum_{m=1}^v |c_{i,m}|^2$ ), the  $SINR$  for UE  $u$  can be expression as:

$$SINR_{i,u} = \frac{\zeta L_{M,0,U} L_{S,0,U} P_l}{\xi P_l + \psi \alpha^2 + \sum_{l=1}^{N_{int}} \theta_{i,l} L_{M,l,u} P_{l,m}} \quad (9)$$

Where  $L_{M,l,u}$  and  $L_{S,o,u}$  are the macro and shadow fading path-losses between UE  $u$  and the attached eNodeB respectively.

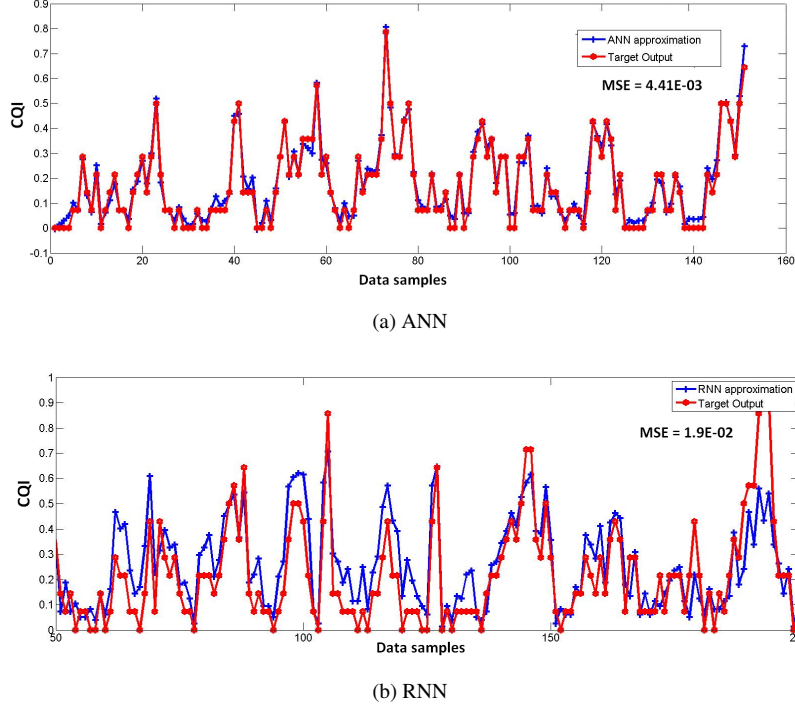


Figure 3: Training for zero feedback delay (Framework 1)

## 7. Cognitive Engine Design 1: AMC as a Classification Problem

The CE design 1 is depicted in Figure 2, which has replaced the state-of-the-art EESM method. The CE is working as a classifier with the aim to select a correct CQI or  $MCS_i$  with respect to different channel state realisations. There are 15 different MCS combinations which can be used to inform eNodeB about the highest MCS that a UE can decode, under the TB error rate constraint. A sample 4-bit CQI table is highlighted in Figure 2, whereas a full CQI index table is presented in [1]. Depending on different channel realizations, the CE selects an accurate CQI indicator.

Prior to the above functioning, a process of training or learning is required. For that, the feeded inputs and outputs to the CE are channel realization matrix ( $X$ ) and ideal MCS matrix ( $Y$ ) respectively. With the feature set  $X$ , label set  $Y$ , and  $n$  training samples  $T = ((x_1, y_1), \dots, (x_n, y_n)) \in (XY)^n$ , the CE creates a mapping  $A: X \rightarrow Y$  from features to labels and predicts the labels for new samples.

To obtain the training dataset, the simulation was run for 1000 TTIs using round robin scheduling and feedback delay of 0 TTI. The CE was trained with RNN-gradient descent and ANN-Levenberg Marquardt. Matlab was used for training and validation of neural networks, where the learning rate for both training algorithms was set to 0.01.

### 7.1. CE Training

For training, different number of neurons, hidden layers and epochs were tried. The best performed RNN and ANN structures were 1 hidden layer with 10 neurons and 1 hidden layer with 20 neurons. Figure 3 depicts the training performance of both ANN and RNN. The least MSE achieved by ANN was 4.41E-03 in 9 iterations and 2s and RNN achieved the MSE of 1.9E-02 in 4 iterations and 15 sec.

### 7.2. CE Testing

#### 7.2.1. Case-1: Throughput Performance in Trained Scenario

In Case-1, the CEs were tested in the environment for which they were trained. Figure 4 illustrates the system throughput comparison among EESM, ANN, and RNN. It can be seen that the performance of ANN is slightly

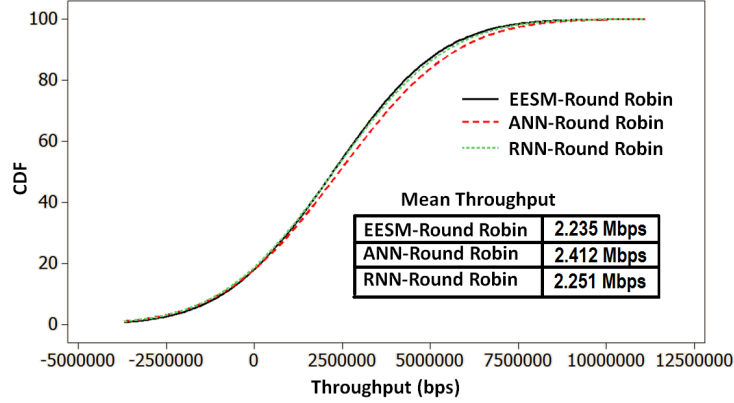


Figure 4: System throughput comparison in trained scenario

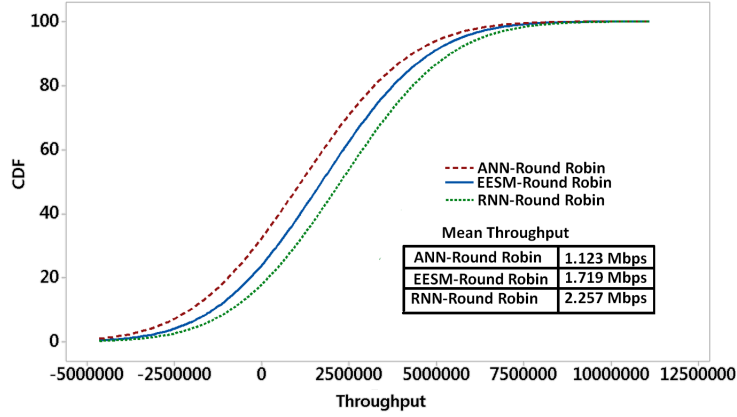


Figure 5: System throughput comparison in untrained scenario

better than RNN and EESM. However, ANN could only perform well for the trained case, which is illustrated in the subsequent section.

#### 7.2.2. Case-2: Throughput Performance in Untrained Scenario (Test for Long-Term Learning)

In this case, the CEs were assumed to be working on a critical mission, where they do not have a privilege of retraining upon extreme propagation environment change. For such cases, long-term learning capability is essential which enables radio to adapt optimal radio parameters in a completely unknown scenario. To investigate the long-term learning, both ANN and RNN models were trained on zero TTI feedback delay dataset but tested in 3 TTIs feedback delay environment. Figure 5 shows the comparison of ANN and RNN with EESM in terms of achieved system throughput in untrained scenario. Figure 6 illustrates the experienced BLER by individual UEs and their achieved throughput respectively. It is to be noted that for 50 TTIs, RNN based UEs consistently experienced the low BLER as compared to high BLER experienced by the ANN based UEs. Moreover, the throughput improvement in RNN based AMC over ANN is also apparent.

#### 7.2.3. Decision Making Speed at Run-Time

In real-time CE applications, fast decision making means that CE can respond quickly upon severe wireless environment change. As a core optimization algorithm, we not only require the decision accuracy but also the response speed. In training phase, the performance of ANN was found to be faster but during run-time the RNN-GD outperformed ANN-LM in total calculation time (decision making speed). For 50 TTIs, ANN took 85.17212 sec to finish the

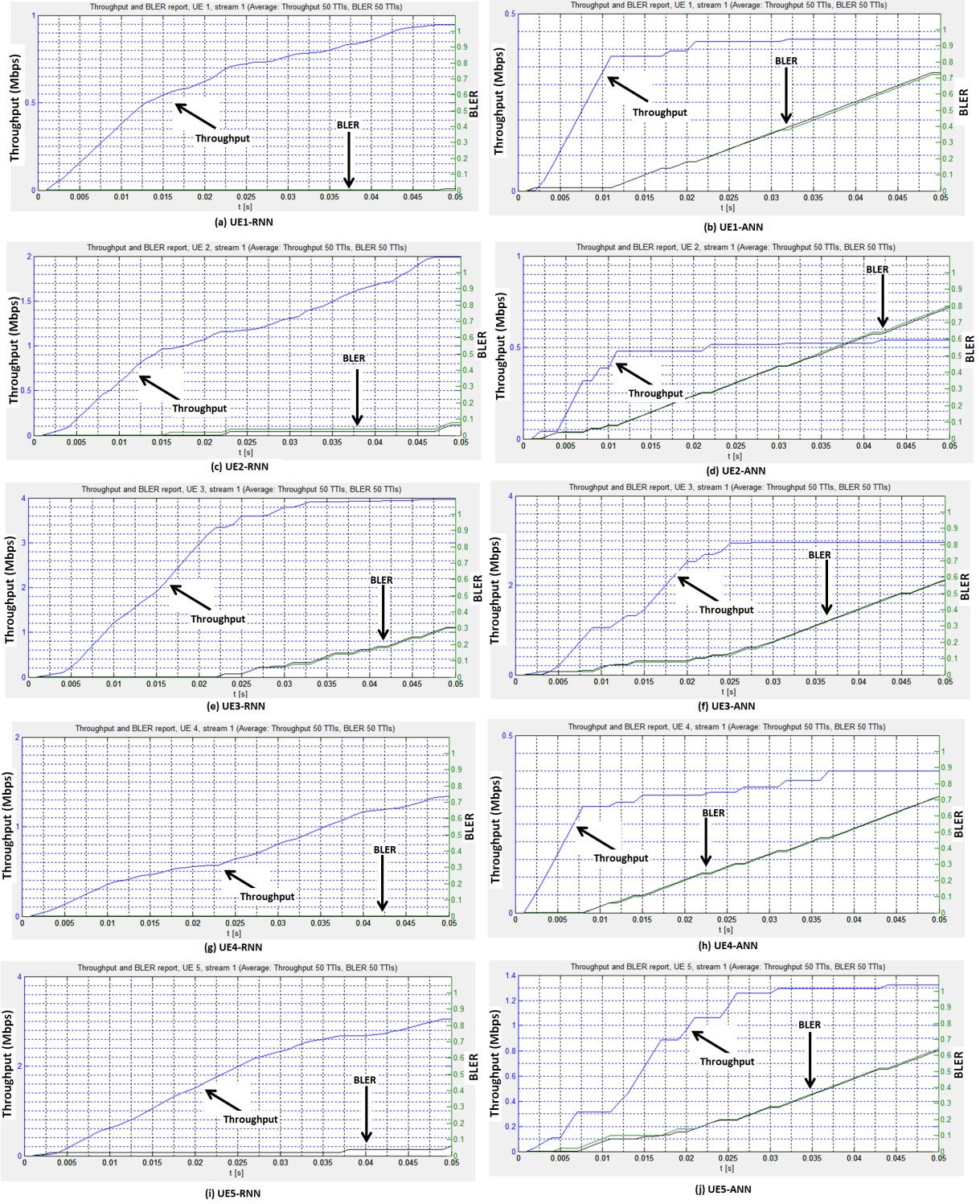


Figure 6: Users experienced BLER in untrained scenario (ANN vs. RNN (framework 1))

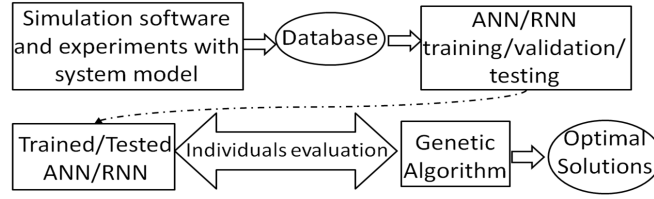


Figure 7: Optimization framework: the data is first collected for learning which helps GA based reasoning process to make optimal decisions

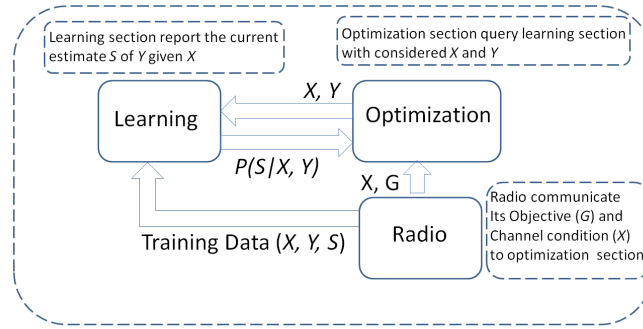


Figure 8: Cognitive engine operation

simulation i.e. 1.70 sec per TTI. In contrast, RNN took just 3.93 sec at the rate of 0.07 sec. RNNs fast decision making during run-time is mainly because of its 3-level architecture in which the computation of output during run-time is extremely fast. The RNNs non-linear system equations computes the output values of input neurons directly from its input and similarly for hidden neurons output. Moreover, in RNN model, the neurons are represented as an integer rather than as a binary variable, which provides more detailed state representation [20]. In contrast, in ANN, the use of non-linear transfer function or activation function such as hyperbolic tangent and sigmoid takes more calculation time as compared to RNN. Furthermore, the hardware/software implementation of ANN is difficult as compared to RNN. For more details and mathematical proofs, see [29].

## 8. Cognitive Engine Design 2: AMC as an Optimization Problem

This section presents an RNN+GA based hybrid CE design. The proposed framework has modelled the MCS selection process as an optimization problem with the aim to maximize the throughput as opposed to maximize the probability of correct classification.

Figures 7 and 8 illustrates the integration of RNN based learning with GA based reasoning. The optimization framework is summarized in Figure 7 where the data is first collected for learning which helps GA based reasoning process to make optimal decisions. A simplified cognitive operation is illustrated in Figure 8, where the decision is to be made based on given environment conditions and current radio objective. The learning module observes the channel  $X$  and estimates the performance  $S$  given radio configuration  $Y$ . The vectors  $X$ ,  $Y$ , and  $S$  are the training parameters coming from the radio. The radio communicates with optimizer the required objectives and current CQI. The optimizer then queries the learning module with considered  $X$  and  $Y$ . The learning section returns the approximate performance of considered  $X$  and  $Y$  i.e.  $P(S|X, Y)$ . Based on this report, the optimization section decides the optimal parameters. In this particular case, the RNN based learning agent is able to characterize the performances of different possible modulation and coding combinations in different channel conditions.

The information which is available to the cognitive controller can be classified into three categories: environmental measurements (external factors effecting the reliability of communication), configuration parameters (tuning knobs

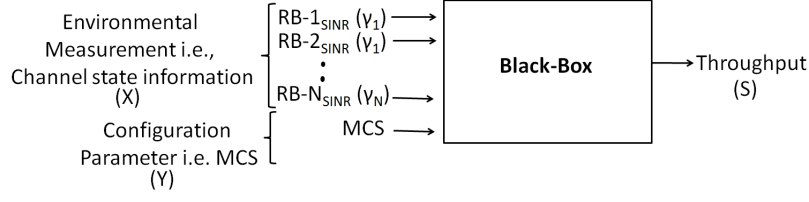
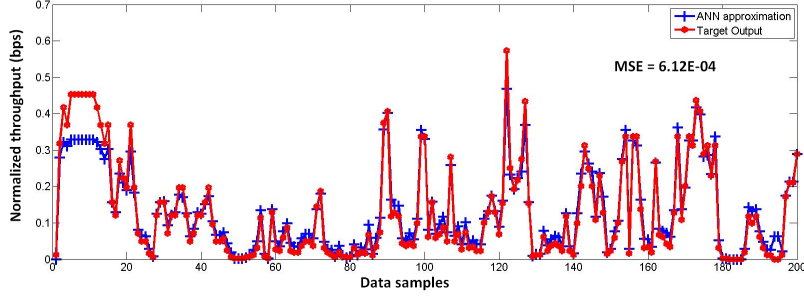
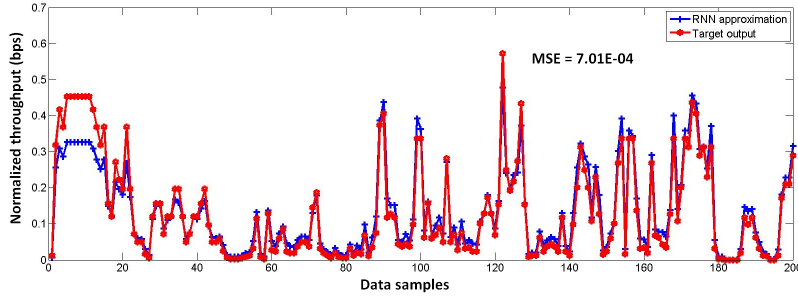


Figure 9: Cognitive Engine Design 2: Learning model



(a) ANN



(b) RNN

Figure 10: ANN vs. RNN training for zero feedback delay (Framework 2)

which can be changed in an optimal way to achieve desired performance), and the performance metric. Based on how different configuration parameters and environmental measurements are affecting the system performance, we fed channel realizations and MCSs as an input to RNN black-box and throughput as an output. Figure 9 depicts the designed learning model where the considered environmental measurements, configuration parameter, and performance metric are also apparent.

### 8.1. Training and testing for different feedback delays

The CQI feedback delay occurs due to the signalling and computation time. This delay directly affects the communication performance in terms of throughput loss, especially in high mobility scenarios. It is desirable to reduce this delay as much as possible, because the channel state significantly changes over the short period of time. In this sub-section, the performance of developed optimization framework is investigated under different feedback delays, ranging from 0 to 3 TTIs.

#### 8.1.1. CQI Feedback Delay=0

In this subsection, the feedback delay was assumed to be zero. The training performance for zero delay is illustrated in Figure 10, where the achieved MSE for both ANN and RNN is approximately same. However, in terms of system throughput, RNN slightly outperformed ANN in achieving more system throughput, as shown in Figure

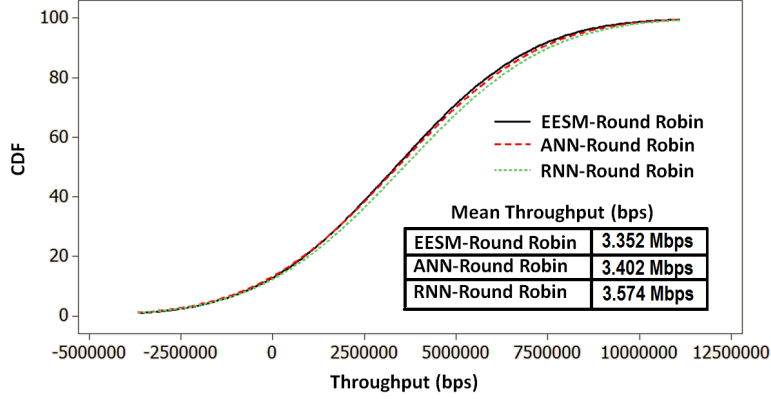


Figure 11: System throughput comparison for zero feedback delay (Framework 2)

Table 2: CQI feedback delay=0: 5%-tile, 25%-tile, and 50%-tile user throughput (Mbps)

Methods	5%-tile	25%-tile	50%-tile
EESM	80.5 kbps	1.560 Mbps	2.221 Mbps
ANN	74.5 kbps	1.328 Mbps	2.221 Mbps
RNN	80.5 kbps	1.560 Mbps	2.221 Mbps

11. Table 2 illustrates the  $5^{th}$ ,  $25^{th}$ , and  $50^{th}$ -tile user throughput in order to compare the achieved coverage and capacity. It is to be noted that RNN has achieved exactly same performance as compared to EESM, whereas ANN achieved 7% less coverage and 15% less user throughput (25%-tile).

## 8.2. Decision Making Speed At Run-Time

The decision making speed of ANN+GA based optimization framework found to be extremely slow. The ANN+GA based optimization framework took 18.65s per decision. In comparison, RNN+GA based optimization framework took 0.61s per decision. It is to be noted that GA with ANN took much longer time to converge than with RNN, which is unreasonable. Therefore, RNN is the optimal choice for the proposed optimization approach. Hence, in rest of the paper, we have only compared the performance of RNN with EESM.

### 8.2.1. CQI Feedback Delay = 1/2/3 TTI

For 1/2/3 TTI feedback delays, the training performance of RNN was found to be appropriately same as in sub-section 8.1.1. Therefore, in this sub-section, only testing performance is reported. Tables 3, 4, and 5 are illustrating the  $5^{th}$ ,  $25^{th}$ , and  $50^{th}$ -tile user throughput. Figure 12 illustrates the system throughput comparison between RNN and EESM for different feedback delays. It can be seen that RNN has significantly outperformed EESM in both coverage and capacity. It is also to be noted that, as the delay increased to 1, 2, and 3 TTIs, the capacity and coverage decreased proportionally. However, RNN handled the delay in a much better way as compared to EESM, which is apparent in Tables 3, 4, 5 and Figure 12. In addition, Figure 13 depicts the average system throughput comparison between RNN and EESM for three different scheduling algorithms at CQI feedback delay of 3 TTIs. It can be seen that the RNN+GA based proportional fair scheduling achieved the highest mean throughput.

Table 3: CQI feedback delay=1 TTI: 5%-tile, 25%-tile, and 50%-tile user throughput (bps)

Methods	5%-tile	25%-tile	50%-tile
EESM	0	0.236 Mbps	2.046 Mbps
RNN	60 Kbps	1.560 Mbps	2.221 Mbps



Table 4: CQI feedback delay=2 TTIs: 5%-tile, 25%-tile, and 50%-tile user throughput

Methods	5%-tile	25%-tile	50%-tile
EESM	0	15 Kbps	1.684 Mbps
RNN	60 kbps	1.56 Mbps	2.221 Mbps

Table 5: CQI feedback delay=3 TTIs: 5%-tile, 25%-tile, and 50%-tile user throughput

Methods	5%-tile	25%-tile	50%-tile
EESM	0	0	0.309 Mbps
RNN	0	0.150 Mbps	1.092 Mbps

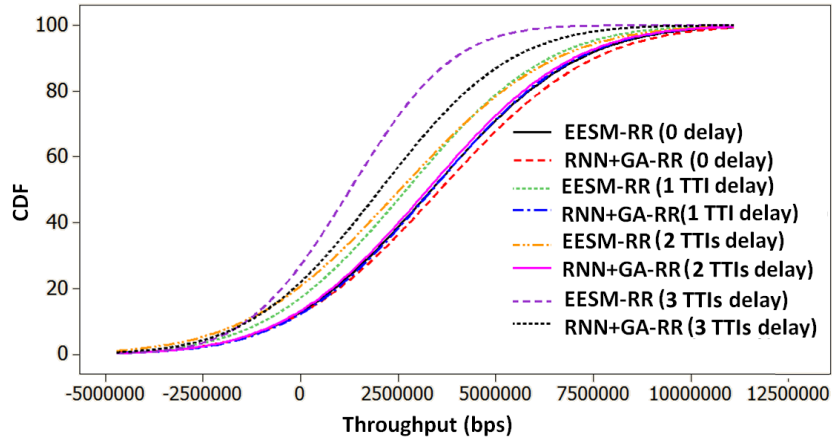


Figure 12: RNN vs. EESM system throughput for 1, 2, and 3 TTIs feedback delay

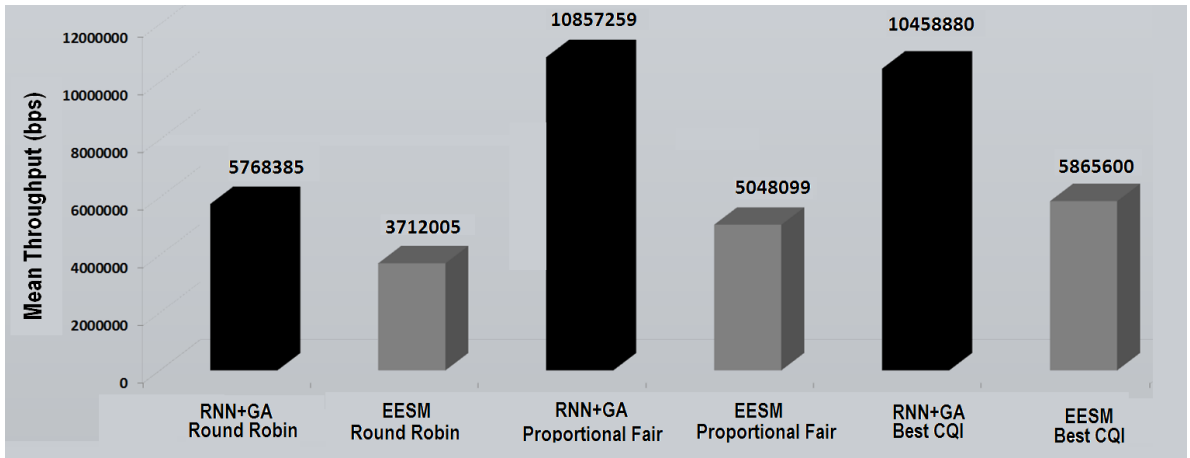


Figure 13: RNN vs. EESM average system throughput for different scheduling algorithms



## 9. Conclusion

In this paper, a novel AMC scheme based on RNN was presented to improve link adaptation in LTE downlink systems. The performance of RNN was compared with ANN and state-of-the-art EESM method, in terms of essential CE design requirements and capacity and coverage optimization. The simulation results revealed that the RNN based CE is capable to concurrently achieve long-term learning, fast decision making, and less computational complexity as compared to ANN. Moreover, RNN+GA based optimization framework can provide significant improvement up-to 100% in both capacity and coverage as compared to EESM. In future, the proposed approach will be used with the mutual information based exponential SNR mapping algorithm under different MIMO transmission modes such as transmit diversity and closed loop spatial multiplexing.

## References

- [1] 3GPP. (2009) 3gpp tr 36.942 v8.2.0 (2009-07). [Online]. Available: [http://www.etsi.org/deliver/etsi\\_tr/136900\\_136999/136942/08.02.00.60/tr\\_136942v080200p.pdf](http://www.etsi.org/deliver/etsi_tr/136900_136999/136942/08.02.00.60/tr_136942v080200p.pdf)
- [2] R. Sandanalakshmi, T. Palanivelu, and K. Manivannan, "Effective snr mapping for link error prediction in ofdm based systems," in *Information and Communication Technology in Electrical Sciences (ICTES 2007)*, 2007. *ICTES. IET-UK International Conference on*. IET, 2007, pp. 684–687.
- [3] E. Tuomaala and H. Wang, "Effective sinr approach of link to system mapping in ofdm/multi-carrier mobile network," in *Mobile Technology, Applications and Systems, 2005 2nd International Conference on*. IEEE, 2005, pp. 5–pp.
- [4] M. Döttling, "Assessment of advanced beamforming and mimo technologies," *European Commission, Project Deliverable IST-WINNER (2003-507581)*, 2005.
- [5] S. N. Donthi and N. B. Mehta, "An accurate model for eesm and its application to analysis of cqi feedback schemes and scheduling in lte," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 10, pp. 3436–3448, 2011.
- [6] J. Francis and N. B. Mehta, "Eesm-based link adaptation in point-to-point and multi-cell ofdm systems: Modeling and analysis," *Wireless Communications, IEEE Transactions on*, vol. 13, no. 1, pp. 407–417, 2014.
- [7] R. C. Daniels, C. M. Caramanis, and R. W. Heath, "Adaptation in convolutionally coded mimo-ofdm wireless systems through supervised learning and snr ordering," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 1, pp. 114–126, 2010.
- [8] H. Yigit and A. Kavak, "Adaptation using neural network in frequency selective mimo-ofdm systems," in *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*. IEEE, 2010, pp. 390–394.
- [9] R. Daniels and R. W. Heath Jr, "Online adaptive modulation and coding with support vector machines," in *Wireless Conference (EW), 2010 European*. IEEE, 2010, pp. 718–724.
- [10] S. Yun and C. Caramanis, "Reinforcement learning for link adaptation in mimo-ofdm wireless systems," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–5.
- [11] Adeel, H. Larijani, and A. Ahmadinia, "Random neural network based cognitive-enodeb deployment in lte uplink," in *Global Telecommunications Conference, 2015. GLOBECOM 2015. IEEE*, (accepted for publication).
- [12] A. Adeel, H. Larijani, and A. Ahmadinia, "Resource management and inter-cell-interference coordination in lte uplink system using random neural network and optimization," *Access, IEEE*, vol. 3, pp. 1963–1979, 2015.
- [13] —, "Random neural network based novel decision making framework for optimized and autonomous power control in lte uplink system," *Physical Communication*, 2015.
- [14] S. Timotheou, "The random neural network: a survey," *The computer journal*, vol. 53, no. 3, pp. 251–267, 2010.
- [15] S. Yun and C. Caramanis, "Multiclass support vector machines for adaptation in mimo-ofdm wireless systems," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 1145–1152.
- [16] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [17] H. I. Volos and R. M. Buehrer, "Cognitive radio engine training," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 11, pp. 3878–3889, 2012.
- [18] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural computation*, vol. 1, no. 4, pp. 502–510, 1989.
- [19] —, "Stability of the random neural network model," *Neural computation*, vol. 2, no. 2, pp. 239–247, 1990.
- [20] —, "Learning in the recurrent random neural network," *Neural Computation*, vol. 5, no. 1, pp. 154–164, 1993.
- [21] A. Adeel, H. Larijani, A. Javed, and A. Ahmadinia, "Critical analysis of learning algorithms in random neural network based cognitive engine for lte systems," in *IEEE Vehicular Technology Conference (VTC), 2015* (accepted for publication).
- [22] Adeel, H. Larijani, A. Javed, and A. Ahmadinia, "Random neural network based power controller for inter-cell interference coordination in lte-ul," in *IEEE ICC 2015 Workshop on Advances in Software Defined and Context Aware Cognitive Networks, 2015*, (accepted for publication).
- [23] L. N. Chakrapani, B. E. Akgul, S. Cheemalavagu, P. Korkmaz, K. V. Palem, and B. Seshasayee, "Ultra-efficient (embedded) soc architectures based on probabilistic cmos (pcmos) technology," in *Proceedings of the conference on Design, automation and test in Europe: Proceedings. European Design and Automation Association*, 2006, pp. 1110–1115.
- [24] L. A. Hey, "Reduced complexity algorithms for cognitive packet network routers," *Computer Communications*, vol. 31, no. 16, pp. 3822–3830, 2008.
- [25] S. Mohamed and G. Rubino, "A study of real-time packet video quality using random neural networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, no. 12, pp. 1071–1083, 2002.

- [26] P. Casas and S. Vaton, "On the use of random neural networks for traffic matrix estimation in large-scale ip networks," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*. ACM, 2010, pp. 326–330.
- [27] J. C. Ikuno, M. Wrulich, and M. Rupp, "System level simulation of lte networks," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*. IEEE, 2010, pp. 1–5.
- [28] C. Mehlführer, J. C. Ikuno, M. Simko, S. Schwarz, M. Wrulich, and M. Rupp, "The vienna lte simulators-enabling reproducibility in wireless communications research," *EURASIP J. Adv. Sig. Proc.*, vol. 2011, p. 29, 2011.
- [29] M. Georgiopoulos, C. Li, and T. Kocak, "Learning in the feed-forward random neural network: A critical review," *Performance Evaluation*, vol. 68, no. 4, pp. 361–384, 2011.